Blind recognition of LDPC code parameters over erroneous channel conditions

Ramabadran, Swaminathan; Madhukumar, A. S.; Wang, Guohua; Ting, Shang Kee

2019

Ramabadran, S., Madhukumar, A. S., Wang, G., & Ting, S. K. (2019). Blind recognition of LDPC code parameters over erroneous channel conditions. IET Signal Processing, 13(1), 86-95. doi:10.1049/iet-spr.2018.5025

https://hdl.handle.net/10356/144700

https://doi.org/10.1049/iet-spr.2018.5025

Blind Recognition of LDPC Code Parameters over Erroneous Channel Conditions

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Swaminathan R[1], A.S.MadhuKumar[1], Wang Guohua[2], and Ting Shang Kee[2]*

[1] *School of Computer Science and Engineering, Nanyang Technological University, Singapore-639798*
[2] *Temasek Laboratories, Nanyang Technological University, Singapore-639798*
* *E-mail: sramabadran@ntu.edu.sg*

Abstract: Blind reconstruction of channel coding parameters plays a significant role in non-cooperative military and spectrum surveillance applications. Further, it also gives additional advantages in applications such as cognitive radio, adaptive modulation and coding, etc. In this paper, blind estimation algorithms are proposed to identify code dimension and codeword length parameters of low-density parity-check (LDPC) codes at the receiver over noisy or erroneous channel conditions assuming a non-cooperative scenario. The proposed algorithms are validated using different test cases. Moreover, the accuracy of estimation of code dimension and codeword length parameters of LDPC codes using the proposed algorithms is also given for different $M-$ary phase-shift keying (PSK) schemes. From the simulation results, it is observed that the accuracy of estimation improves with decrease in code rate, codeword length, and modulation order.

1 Introduction

Forward error correcting (FEC) codes play a vital role in improving the error performance of digital communication and storage systems by correcting the random errors due to noisy channel conditions. Low-density parity-check (LDPC) codes are used in various applications such as Digital video broadcasting-satellite-second generation (DVB-S2) system, IEEE 802.11n, and IEEE 802.11ac Wi-Fi systems, etc. In most of the applications, the receiver has the knowledge of FEC code parameters used for transmission. However, only limited knowledge about the code parameters is available at the receiver in non-cooperative scenarios [1], which exist particularly in military, spectrum surveillance, signals intelligence (SIGINT), and communications intelligence (COMINT) systems. Therefore, it is essential to reconstruct the channel encoder by identifying the code parameters using the intercepted sequences acquired from remote sensing through aircraft and satellite. Further, the blind identification of code parameters is also useful in applications such as adaptive modulation and coding (AMC), cognitive radio, etc. In general, the FEC code parameters are transmitted using control channel to the receiver in AMC-based systems. Hence, the blind reconstruction of channel encoder will help in conserving the channel resources as mentioned in [2]-[4] and improves the spectral efficiency of the AMC-based systems. Note that wireless sensor networks (WSNs) also adopt AMC and blind parameter estimation algorithms help to reduce transmission overheads and total energy consumption of WSNs [3]. Further, the cognitive radio receivers need to comply with the variations in the channel coding scheme in order to decode the message symbols correctly. Thus, the blind identification of FEC code parameters finds potential applications in cognitive radio systems. It is also a costly and a tedious process to design a separate decoding system for every application with the evolution of modern digital communication systems. Hence, it is always mandatory to design an intelligent receiver system which adapts itself to any specific application [5] and [6]. The parameter estimation techniques is also useful in the study of DNA sequences to identify possible error correcting codes in the genetic code as reported in [7] and [8] apart from the mentioned applications.

In [1], the code parameter recognition algorithms were given for different FEC codes considering noiseless scenario. In [5], [6], and [9], the blind estimation algorithms for estimating convolutional code parameters were reported for noisy scenario. In [10] and [11], the blind identification algorithms were proposed for estimating punctured convolutional code parameters together with the puncturing pattern. In addition, the dual code properties for blind parameter estimation of convolutional codes were proposed in [12]. The algorithm for blind estimation of binary cyclic code parameters was proposed in [13]. In [14], the code classification algorithms were proposed to classify the incoming FEC coded symbols among block coded, convolutionally coded, and uncoded symbols. The blind identification techniques for recognizing the codeword length of various non-binary error correcting codes including LDPC codes were reported for noisy environment in [15]. However, the codeword lengths assumed in the proposed work were very small. Further, the blind reconstruction of Reed-Solomon (RS) encoder with and without interleaver was investigated in [16] for noisy scenario. In [17], a novel technique based on Gauss Jordan elimination through pivoting was proposed for the blind recognition of LDPC, turbo, and convolutional code parameters. However, the technique is applicable only to non-erroneous scenario. The code parameter recognition technique based on the average log-likelihood ratio (LLR) of syndrome a posteriori probability was proposed in [2]-[4], [18], and [19] to identify true RS encoder, LDPC encoder, and convolutional encoder. In [20], the channel code parameters within the candidate set were identified based on the average likelihood difference of the parity checks with reduced computational complexity. It was also reported that enhanced performance is observed compared to the LLR-method for 1/2-rate convolutional codes proposed in [4] and [18]. In [21] and [22], the individual recursive systematic convolutional (RSC) code parameters of a turbo code using iterative expectation-maximization and least square methods were estimated, respectively. Besides estimating the FEC code parameters, the blind estimation algorithms for various interleavers such as block and convolutional interleavers were proposed in [23]-[30].

1.1 Motivations

The main motivations of the proposed work are given as follows:

- In [17], a novel method for identifying parameters of LDPC codes was proposed. However, it is limited to noiseless (i.e. non-erroneous) scenario and is suitable only for moderate codeword lengths.

• In prior works [3] and [19], the estimation of true LDPC encoder has been carried out for binary phase-shift keying (BPSK) scheme based on the LLR approach, which is more suitable for AMC-based systems. However, the LLR-based estimation process is not strictly blind, since a predefined LDPC encoder candidate set is assumed to be known at the transmitter and receiver.
• But in practical non-cooperative applications such as military and spectrum surveillance, the candidate set is unknown and hence, it is mandatory to blindly estimate LDPC code parameters before applying the reconstruction process.
• Blind recognition algorithms without assuming a predefined candidate set for LDPC codes considering large codeword lengths for noisy scenario have not been reported to the best of our knowledge.

1.2 Contributions

The main contributions of the proposed work are given as follows:

• In this paper, we propose blind estimation algorithms to identify codeword length n and code dimension k of LDPC codes for noisy channel conditions assuming a non-cooperative scenario.
• Simulation results validating the proposed algorithms are given for various test cases considering $M-$ary phase-shift keying (PSK) scheme, where M denotes the modulation order, for three different codeword lengths (i.e. $n=648$, $n=1296$, and $n=1944$).
• The accuracy of estimation of the proposed algorithms with respect to different signal-to-noise ratio (SNR), codeword length n, and modulation order M values is given along with detailed discussions.

2 LDPC Codes and parameter estimation process

LDPC codes are shown to achieve better error performance compared to turbo codes. It is only a fraction of decibel (dB) away from the Shannon limit of -1.6dB, which is the minimum SNR required to achieve arbitrarily small error probability. Hence, it is being currently used in many communication and storage systems, where high reliability is required. An LDPC code can be specified in terms of parity check matrix H with dimension $(n-k)\times n$ and is defined as the null space of H, which has the following properties [31]:

• Each row of H consists of χ 1's.
• Each column of H consists of γ 1's.
• No two rows or columns of H have more than one 1 in common.
• Both χ and γ are smaller compared to the number of rows of H and n. Therefore, H is a sparse matrix or a low density matrix and hence, the code specified by sparse H matrix is called LDPC code. From first two properties, it is inferred that H matrix has constant row and column weights χ and γ, respectively. In such cases, the LDPC code is called regular LDPC code. If all the columns or rows of H matrix do not have same weight, then the LDPC code is said to be irregular.

The block diagram of the parameter estimation process of LDPC codes is shown in Fig. 1. The binary LDPC encoded bit streams are modulated using $M-$PSK scheme and transmitted over additive white Gaussian noise (AWGN) channel. The code parameters are estimated from the erroneous received binary coded data symbols after demodulation and our discussions are mainly restricted to estimation of k and n alone.

3 Parameter estimation of LDPC codes

The steps for the estimation of k and n for less erroneous channel conditions are given using Algorithm 1. The incoming LDPC encoded binary symbols are reshaped into a matrix S of size $a\times b$, where a and b denote the number of rows and columns of S, respectively. Since S contains coded data symbols, it is called data matrix. The rank and rank ratio of S are calculated using Gauss elimination process. It is to be noted that the ratio of the rank of a matrix to the number of columns gives the rank ratio. The rank of a matrix is the number of linearly independent rows/columns of a matrix. It can be obtained by transforming the given matrix into its row/column echelon form using Gauss elimination method. The number of non-zero rows/columns of row/column echelon form gives the rank. In this manuscript, column-wise operation is performed and the column echelon form of S is denoted by F. It is to be noted that similar operations can be performed row-wise as well. From the rank ratio values, codeword length n is estimated by noticing the difference between successive number of columns with same deficient rank ratio values. For instance, if there is only one deficient rank value $\rho(b)$, then n is obtained as mentioned in step 5 of Algorithm 1. Further, code dimension k is estimated by finding the difference between corresponding rank values of rank-deficient columns. However, if there is only one deficient rank value in the search space, then k is estimated as mentioned in step 6 of Algorithm 1. Please note that S is a full rank matrix only if $\text{rank}(S)=\min(a,b)$ or rank ratio is equal to unity. Further, S is a deficient rank matrix only if $\text{rank}(S)<\min(a,b)$ or rank ratio is less than unity.

Algorithm 1: Estimation of k and n for less erroneous channel conditions

Notations: Let a, b, $\rho(b)$, and p denote the number of rows, columns, rank, and rank ratio of data matrix S, respectively. Let F denotes the column echelon form of S. Finally, n_{est} and k_{est} denote the estimate of n and k, respectively.
Assumptions: $a=2\cdot b$ and $b\in[b_{\min},\ b_{\max}]$.
Input: LDPC encoded binary data symbols;
Output: n_{est} and k_{est};
1: **for** $b=b_{min}:b_{max}$ **do**
  2: Reshape the received LDPC encoded data symbols into a matrix S of size $a\times b$;
  3: Convert S into F using Gauss elimination process;
  4: Calculate $\rho(b)$ by observing the number of non-zero columns in F and the rank ratio p is given by $p=\rho(b)/b$;
end
5: Estimate n by observing the difference between successive number of columns with same deficient rank ratio values or if there is only one deficient rank value $\rho(b)$ in the search space, then obtain n as follows: $[n_{\text{est}}]=\underset{b}{\text{argmin}}(\rho(b))$;
6: Obtain k by finding the difference between corresponding rank values of rank-deficient columns or it can be also obtained as follows: $[k_{\text{est}}]=\rho(n_{\text{est}})$;

The reason for rank deficiency in step 5 is explained as follows: Here, systematic block encoding is used to explain the rank deficiency and full rank phenomena for better clarity. But the proposed algorithm also works in the case of non-systematic block encoding. Since LDPC codes belong to the family of linear block codes, the following discussion on deficient and full rank phenomena considering linear block codes can also be extended to LDPC codes. In case of linear block codes, n output block coded data symbols will depend on k input uncoded data symbols [14], [23], and [24]. Similarly, $\alpha\cdot n$ output block coded data symbols will depend on $\alpha\cdot k$ input uncoded symbols. Hence, if b is a multiple of n (i.e. $b=\alpha\cdot n$), then α codewords in a particular row will depend on $\alpha\cdot k$ symbols. This is also applicable to all other rows of S. Further, it has been observed that if $b=\alpha\cdot n$, then the message and parity bits are aligned properly in the same column across all the rows as depicted in Fig. 2(a) and the linear relation is satisfied in all the rows. Therefore, there will exist linear relations between columns in S. After converting S into F using Gauss elimination process, only $\alpha\cdot k$ non-zero or independent columns out of b columns will be observed for the case when $b=\alpha\cdot n$ and the rest of $\alpha\cdot(n-k)$ dependent columns will be eliminated for non-erroneous and less erroneous channel conditions. Hence, rank deficiency will be obtained.

It is also observed that the data and parity symbols of α codewords in all the rows are not aligned properly in the same column for the case when b is not a multiple of n i.e. $b\neq\alpha\cdot n$ as depicted in Fig.

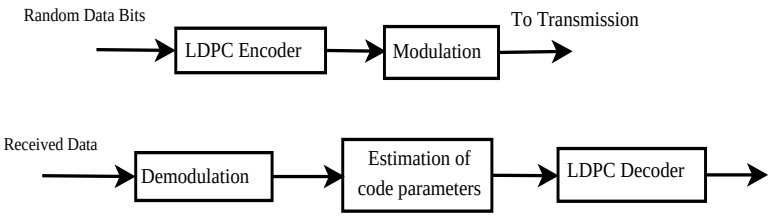


Fig. 1: *Block diagram of parameter estimation process of LDPC codes*

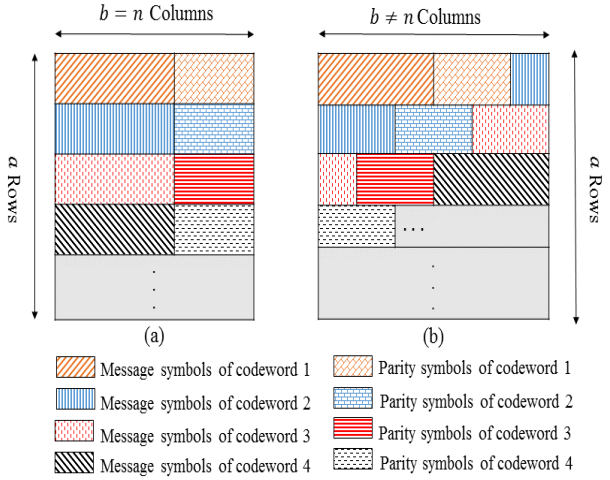


Fig. 2: *Structure of data matrix for the case when (a) $b=n$ and (b) $b\neq n$*

2(b). If the alignment is not proper, then the linear relation in all the rows will be affected, which in turn will lead to disappearance of linear relations between columns in S. Therefore, S will behave like a random matrix without any dependent columns, which will result in full rank. For better understanding, a case study explaining the rank deficiency and full rank phenomena considering linear block code has been discussed.

It is assumed that the input sequence $(t_1, t_2, t_3, t_4, \cdots)$ enters the systematic block encoder $B(7,4)$, where codeword length $n=7$ and code dimension $k=4$, and the corresponding codeword is given by $(t_1, t_2, t_3, t_4, g_1, g_2, g_3, \cdots)$, where g_1, g_2, and g_3 denote the parity symbols corresponding to the input sequence (t_1, t_2, t_3, t_4). Similarly, g_4, g_5, and g_6 denote the parity symbols corresponding to the input sequence (t_5, t_6, t_7, t_8). The incoming message and parity symbols are reshaped into a matrix S of size $a \times b$ for $b=7, 10,$ and, 14 assuming number of rows $a=3$ as shown in Table 1. Please note that one and two complete codewords are identified in all the three rows for $b=7$ and $b=14$, respectively. It is also noticed that the data and parity symbols are also properly aligned in the same column for both the cases. Since a codeword of length n bits depend only on k information bits in case of linear block codes, $\alpha.n$ output bits depend on $\alpha.k$ input bits. Because of the proper alignment of data and parity bits, the linear relation is satisfied in all the rows and this implies that the columns in S are also linearly related. Hence, after converting S into F using Gauss elimination process, only $\alpha \cdot k=4$ non-zero or independent columns will be observed for $b=7$ and $\alpha \cdot k=8$ independent columns will be observed for $b=14$. It is to be noted that $\alpha=1$ for $b=7$ and $\alpha=2$ for $b=14$. As already mentioned, the number of independent columns of a matrix or the number of non-zero columns in a column echelon form of a matrix gives the rank of the matrix. The corresponding rank values obtained for $b=7$ and 14 well agree with the rank values shown in Fig. 3(a). If $b\neq\alpha.n$, then it is noticed from Table 1 that the data and parity bits are not aligned properly in the same column (refer to $b=10$ case). Due to improper alignment, linear relations will not exist between the columns in S and it is equivalent to a random matrix. This will result in full rank as shown in Fig. 3(a).

Since $b=\alpha \cdot n$ case gives deficient rank, we obtain codeword length parameter by noticing the difference between successive number of columns with deficient rank ratio values. On the other hand, if there is only one deficient rank value in the search space, then we obtain codeword length parameter from the number of columns b, which minimizes $\rho(b)$.

In a nutshell, let S be a data matrix of coded symbols with b columns. If $b=\alpha \cdot n$, where α is a positive integer, then rank $\rho(b)$ and rank ratio p of S are given by

$$\begin{aligned} \rho(b) &= \alpha \cdot k < b\,, \\ p &= \frac{\rho(b)}{b} = r\,, \end{aligned} \tag{1}$$

where r is the code rate of LDPC codes.

Let $b=\alpha \cdot n$ and $b'=(\alpha+1)\cdot n$ denote two successive columns with rank deficiency or same deficient rank ratio values. From $b'-b$, the codeword length can be identified as follows:

$$n_{\text{est}} = ((\alpha+1)\cdot n) - (\alpha \cdot n)\,. \tag{2}$$

Further, the difference between corresponding rank values of rank-deficient columns b and b' gives the estimate of code dimension k as follows:

$$k_{\text{est}} = ((\alpha+1)\cdot k) - (\alpha \cdot k)\,. \tag{3}$$

It has been observed that for non-erroneous and less erroneous channel conditions, all the dependent columns will be converted into all-zero columns using Gauss elimination process and hence, deficient rank values are obtained. For example, the rank versus number

Table 1 Block code $B(7,4)$ reshaped into a data matrix S of size $3\times b$

$b=7$	t_1	t_2	t_3	t_4	g_1	g_2	g_3							
	t_5	t_6	t_7	t_8	g_4	g_5	g_6							
	t_9	t_{10}	t_{11}	t_{12}	g_7	g_8	g_9							
$b=10$	t_1	t_2	t_3	t_4	g_1	g_2	g_3	t_5	t_6	t_7				
	t_8	g_4	g_5	g_6	t_9	t_{10}	t_{11}	t_{12}	g_7	g_8				
	g_9	t_{13}	t_{14}	t_{15}	t_{16}	g_{10}	g_{11}	g_{12}	t_{17}	t_{18}				
$b=14$	t_1	t_2	t_3	t_4	g_1	g_2	g_3	t_5	t_6	t_7	t_8	g_4	g_5	g_6
	t_9	t_{10}	t_{11}	t_{12}	g_7	g_8	g_9	t_{13}	t_{14}	t_{15}	t_{16}	g_{10}	g_{11}	g_{12}
	t_{17}	t_{18}	t_{19}	t_{20}	g_{13}	g_{14}	g_{15}	t_{21}	t_{22}	t_{23}	t_{24}	g_{16}	g_{17}	g_{18}

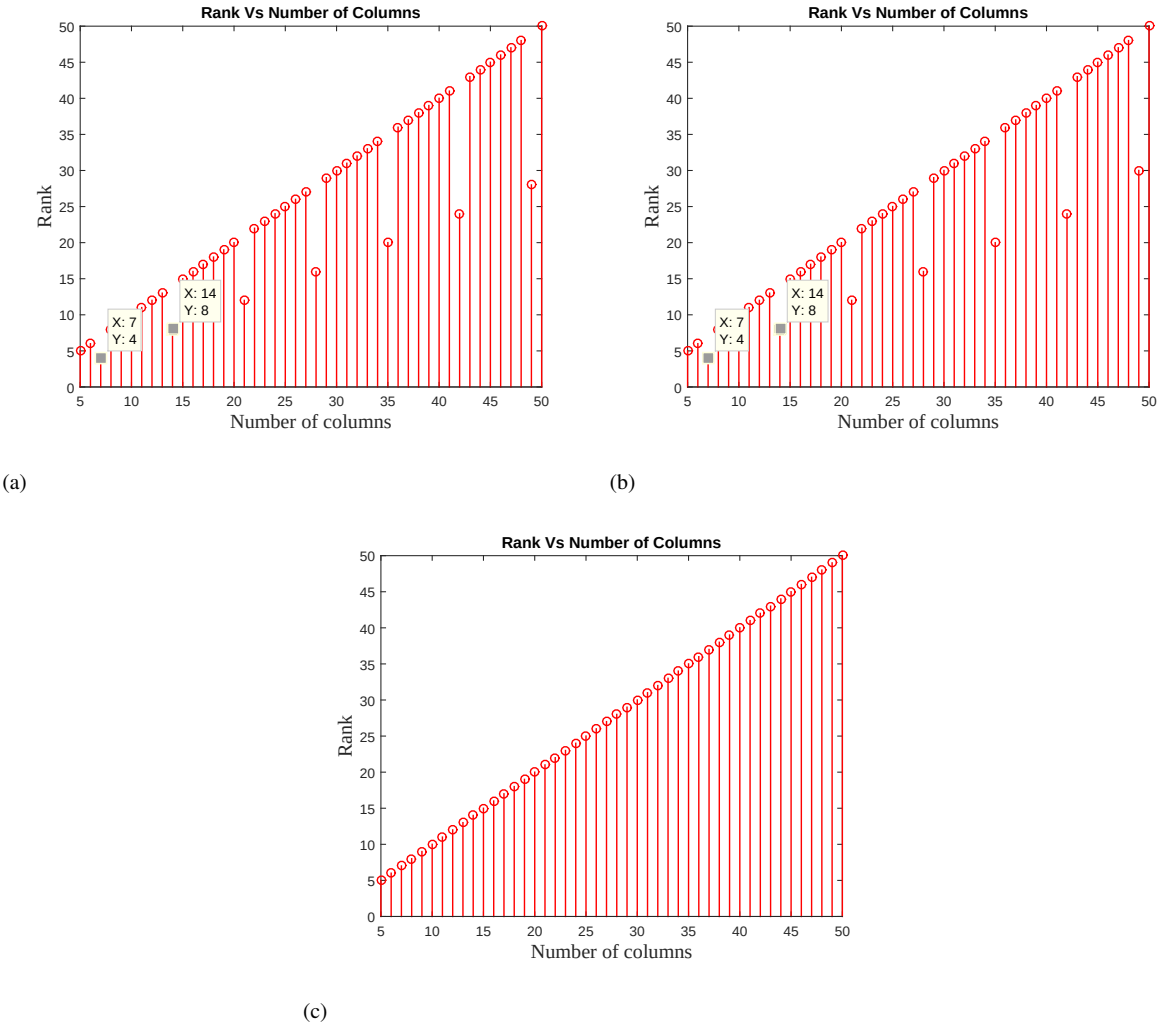


Fig. 3: *(a) Rank $\rho(b)$ versus number of columns b for $B(7,4)$ considering non-erroneous scenario (b) $\rho(b)$ versus b for $B(7,4)$ considering bit error rate (BER) = 5×10^{-4} (c) $\rho(b)$ versus b for $B(7,4)$ considering BER = 5×10^{-2}*

of columns plot is shown in Fig. 3(b) for $B(7,4)$ considering bit error rate (BER) = 5×10^{-4}. It is noticed that the rank values exactly match with the values obtained for non-erroneous scenario (refer to Fig. 3(a)). Therefore, Algorithm 1 is not only applicable to non-erroneous channel conditions. It is also applicable to high-SNR or less erroneous channel conditions. However, Algorithm 1 fails in case of low-SNR or more erroneous channel conditions [14]. This is mainly due to the presence of transmission errors or white noise. As the transmission errors increases, the linear independence among rows/columns of a deficient rank matrix also increases. When the noise level exceeds a threshold SNR or BER value, the received data matrix S will not have any dependent columns and will behave like a random matrix. Hence, full rank will be obtained irrespective of b. For example, in Fig. 3(c), the variation of rank values with respect to the number of columns is shown for $B(7,4)$ considering the case when BER $= 5\times10^{-2}$. It is observed that full rank is obtained irrespective of b. Thus, Algorithm 1 is not suitable for more erroneous channel conditions to estimate n and k. However, it is noticed that the dependent columns in S will have more number of zero elements in F compared to independent columns. Therefore, we modify Algorithm 1 for more erroneous scenario and the code parameters are estimated based on zero-mean-ratio $\mu(b)$ of F over N iterations using Algorithm 2

In Algorithm 2, the zero-mean-ratio value of column echelon form F (i.e. $\mu(b)$) is calculated instead of $\rho(b)$. To evaluate $\mu(b)$, the mean value of the number of zeros in each column of F_j, which is denoted by $\omega_j(c)$, is first calculated and the same is repeated for N iterations, where $c\in\{1,2,...,b\}$ and j denotes the iteration number. It is to be noted that the column echelon form of data matrix S_j is

Algorithm 2: Estimation of n for more erroneous channel conditions

Notations: Let $\mu(b)$ denotes the zero-mean-ratio, F_j denotes the column echelon form of data matrix S_j, j denotes the iteration number, $\omega_j(c)$ denotes the mean value of the number of zeros in c^{th} column of F_j, $data(\cdot)$ refers to an array of incoming LDPC coded bits, N_{frame} denotes the number of frames, num_data indicates the number of elements in the array $data(\cdot)$, and N refers to the number of iterations;
Assumptions: $a=2\cdot b$ and $N_{\text{frame}}=\text{floor}(num\_data/b)$ and $N=N_{\text{frame}}-a+1$;
Input: LDPC encoded binary data symbols;
Output: n_{est};
1: **for** $b=b_{min}:b_{max}$ **do**
  2: **for** $j=1:N$ **do**
    3: $R_j=data(1+(j-1)b\ :\ ba+(j-1)b)$;
    4: Reshape R_j into a matrix S_j of size $a\times b$;
    5: Convert S_j into F_j using Gauss elimination process;
    6: Compute $\omega_j(c)$ in each column of F_j, where $c\in\{1,2,...,b\}$;
    7: Form a row matrix $A_j=[\omega_j(1)\ :\ \omega_j(b)]$;
  end
  8: Accumulate all the row matrices into a single matrix A of size $N\times b$, where $A=[A_1\ ;\ A_2\ ;\ A_3\ ;\ \cdots A_N]$;
  9: Compute $B=mean(A)$, where $B=[\sigma(1)\ :\ \sigma(b)]$ and $\sigma(c)=\frac{\sum_{j=1}^{N}\omega_j(c)}{N}$;
  10: Compute $\mu(b)=\frac{\sum_{c=1}^{b}\sigma(c)}{b}$;
end
11: Estimate n by observing the difference between successive number of columns with higher values of $\mu(b)$ or obtain n as follows: $[n_{\text{est}}]=\underset{b}{\text{argmax}}(\mu(b))$;

denoted by F_j. The calculated values for all N iterations are accumulated into a single matrix A of size $N\times b$. Now the mean of A is evaluated and the resultant row vector of size $1\times b$ is denoted by B, where $B=[\sigma(1)\ :\ \sigma(b)]$ and $\sigma(c)=\frac{\sum_{j=1}^{N}\omega_j(c)}{N}$. From $\sigma(c)$, the zero-mean-ratio is calculated as follows: $\mu(b)=\frac{\sum_{c=1}^{b}\sigma(c)}{b}$. For example, if c^{th} column in S is dependent and if c^{th} column in F is not an all-zero column due to erroneous bits, then it is intuitive that the mean value of the number of zeros in c^{th} dependent column will be higher compared to the independent columns. Hence, zero-mean-ratio $\mu(b)$ of column echelon form F will be higher for rank deficient data matrix compared to full rank data matrix. Note that the full rank matrix will have zero and non-zero elements with equally-likely probability. Thus, the rank deficient matrix over more erroneous channel conditions is identified based on the number of zero elements in F instead of number of non-zero columns. In this context, Algorithm 1 is modified and the code parameters are estimated based on zero-mean-ratio of F. Note that S is a rank deficient matrix only for the case when b is a multiple of n i.e. $b=\alpha\cdot n$. Therefore, the codeword length n is identified by observing the difference between successive number of columns with higher values of $\mu(b)$. Otherwise, the value of b which maximizes $\mu(b)$ is recognized as codeword length of LDPC code as mentioned in step 11 of Algorithm 2. Since deficient rank values cannot be obtained precisely using Algorithm 2 due to more number of erroneous bits, k cannot be estimated successfully. However, n alone can be estimated for more erroneous channel conditions using $\mu(b)$.

4 Simulation results and discussions

To estimate n and k using the variations of $\mu(b)$ and $\rho(b)$ with respect to b, three different codeword lengths (i.e. $n=648$, $n=1296$, and $n=1944$) are considered in our simulation study. The LDPC codes assumed are irregular codes and they are prominently used in IEEE 802.11-2012 and IEEE 802.11n Wi-Fi standards.

4.1 Parameter estimation using Algorithm 1 and 2

In Fig. 4(a), the variation of $\rho(b)$ with respect to b is shown for LDPC encoder with code parameters $n=648$, $k=324$, and code rate $r=\frac{1}{2}$ assuming 8-PSK modulation scheme, SNR = 13 dB, and $a=2\cdot b$. From the plot, it is observed that for $b=648$, rank deficiency is obtained and the deficient rank value is given by $\rho(b)=324$. For rest of the values of b, full rank is obtained. Hence, by observing the number of columns b with rank deficiency and associated rank value $\rho(b)$, n and k are estimated, respectively, for LDPC encoder using Algorithm 1. In Fig. 4(b), the variation of $\rho(b)$ with respect to b is shown for the same case assuming SNR = 12 dB. From the plot, it is inferred that the rank deficiency is obtained when $b=648$. Therefore, n is successfully identified for SNR = 12 dB using Algorithm 1. However, the deficient rank value does not give the correct estimate of k due to more number of erroneous bits.

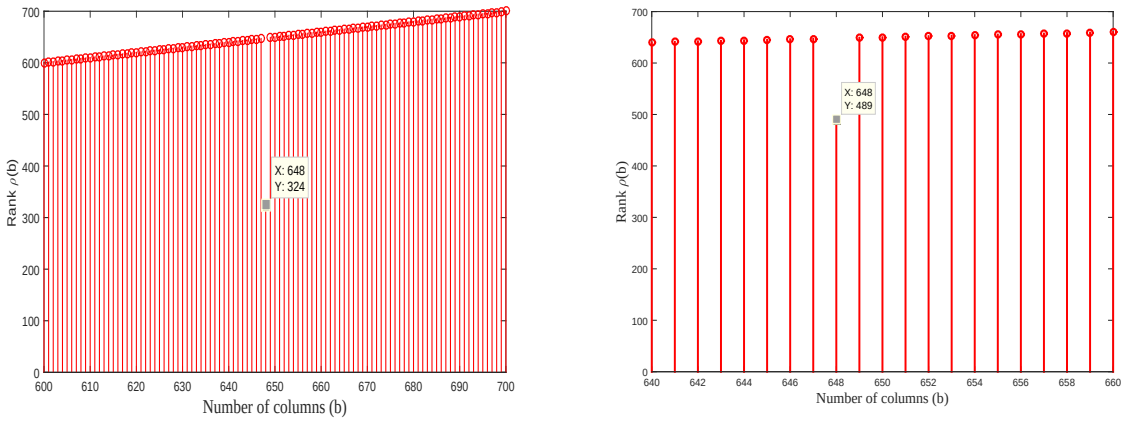
It is to be noted that n can be identified successfully for SNR $\geq$ 12 dB using Algorithm 1. When the SNR is further decreased to 11 dB, it is noticed from Fig. 5(a) that both n and k cannot be estimated using Algorithm 1 based on the rank deficiency approach. Because, full rank values are obtained irrespective of b. Therefore, we have calculated the zero-mean-ratio of F i.e. $\mu(b)$ instead of $\rho(b)$ according to Algorithm 2. From Fig. 5(b), it is observed that $\mu(b)$ attains maximum at $b=648$ due to rank deficiency. Hence, n is successfully identified based on $\mu(b)$ for SNR = 11 dB. However, when SNR $<$ 11 dB, n cannot be recognized successfully based on Algorithm 2.

In Fig. 6(a), the variation of $\rho(b)$ with respect to b is shown for LDPC encoder assuming $r=\frac{1}{2}$, $n=1296$, $k=648$, $M=4$, SNR = 12 dB, and $a=2\cdot b$. It is observed from the figure that rank deficiency is obtained for $b=1296$ and the deficient rank value is given by $\rho(b)=648$. It is also noticed that full rank is obtained for rest of the values of b. Therefore, n and k are estimated from the number of columns b with rank deficiency and $\rho(b)$, respectively, using Algorithm 1. In Fig. 6(b), the variation of $\rho(b)$ with respect to b is shown for SNR = 11 dB. It is inferred that the rank deficiency is obtained when $b=1296$ and hence, n is successfully identified. But $\rho(b)$ at $b=1296$ does not give the correct estimate of k due to more number of erroneous bits. Further, n can be identified successfully for SNR $\geq$ 11 dB and when the SNR is decreased to 10 dB, both n and k cannot be estimated using Algorithm 1 due to full rank values irrespective of b. Therefore, $\mu(b)$ is calculated according to Algorithm 2 and it is observed from Fig. 6(c) that $\mu(b)$ attains maximum at $b=1296$ due to rank deficiency. Hence, n is successfully identified based on $\mu(b)$ according to Algorithm 2 for SNR = 10 dB. However, when SNR further decreases, Algorithm 2 fails to recognize n.

From Fig. 7(a), Fig. 7(b), Fig. 8(a), and Fig. 8(b), it is inferred that $n=1944$ and $k=972$ are identified successfully for SNR $\geq$ 11 dB considering BPSK scheme. Further, n alone can be correctly identified for SNR $\geq$ 9 dB and SNR $\geq$ 8 dB using Algorithm 1 (based on $\rho(b)$) and Algorithm 2 (based on $\mu(b)$), respectively. Finally, when SNR $<$ 8 dB, Algorithm 2 fails to identify n.

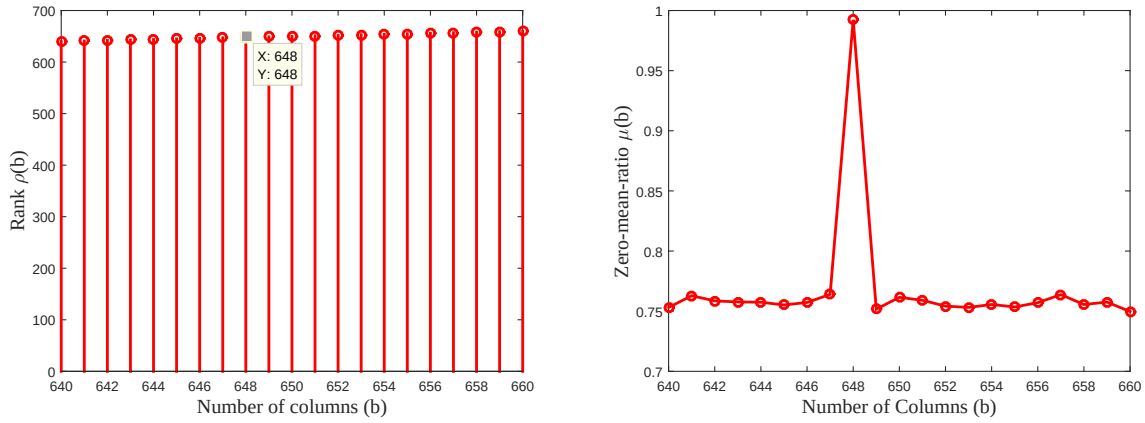
4.2 Accuracy of estimation of code parameters

In Table 2, Table 3, and Table 4, the accuracy of estimation of number of dependent columns (i.e. $n-k$) based on Algorithm 1 considering $b=n$ is shown for $n=648$, $n=1296$, and $n=1944$ cases, respectively. The tabulated entries are the estimated number of dependent columns out of actual number of dependent columns $n-k$ and the percentage of accuracy is given within the braces. Note that there will be k independent and $n-k$ dependent columns for $b=n$ over non-erroneous or ideal channel conditions. We consider BPSK and quadrature phase-shift keying (QPSK) modulation schemes assuming $r=5/6$ and $r=1/4$. For instance, if $r=5/6$ and $n=648$, then $n-k=108$. Similarly, $n-k=486$ for $r=1/4$ and $n=648$. It is to be noted that if all the dependent columns are identified, then n and k can be estimated accurately. From Table 2, it



(a) (b)

Fig. 4: *(a) Rank Vs Number of columns considering* $r = 1/2$, $n = 648$, $k = 324$, $M = 8$, *SNR = 13 dB, and* $a = 2 \cdot b$ *(b) Rank Vs Number of columns considering* $r = 1/2$, $n = 648$, $k = 324$, $M = 8$, *SNR = 12 dB, and* $a = 2 \cdot b$



(a) (b)

Fig. 5: *(a) Rank Vs Number of columns considering* $r = 1/2$, $n = 648$, $k = 324$, $M = 8$, *SNR = 11 dB, and* $a = 2 \cdot b$ *(b) Zero-mean-ratio Vs Number of columns considering* $r = 1/2$, $n = 648$, $k = 324$, $M = 8$, *SNR = 11 dB, and* $a = 2 \cdot b$

Table 2 Accuracy of estimation of dependent columns $n - k$ based on Algorithm 1 assuming $n = 648$

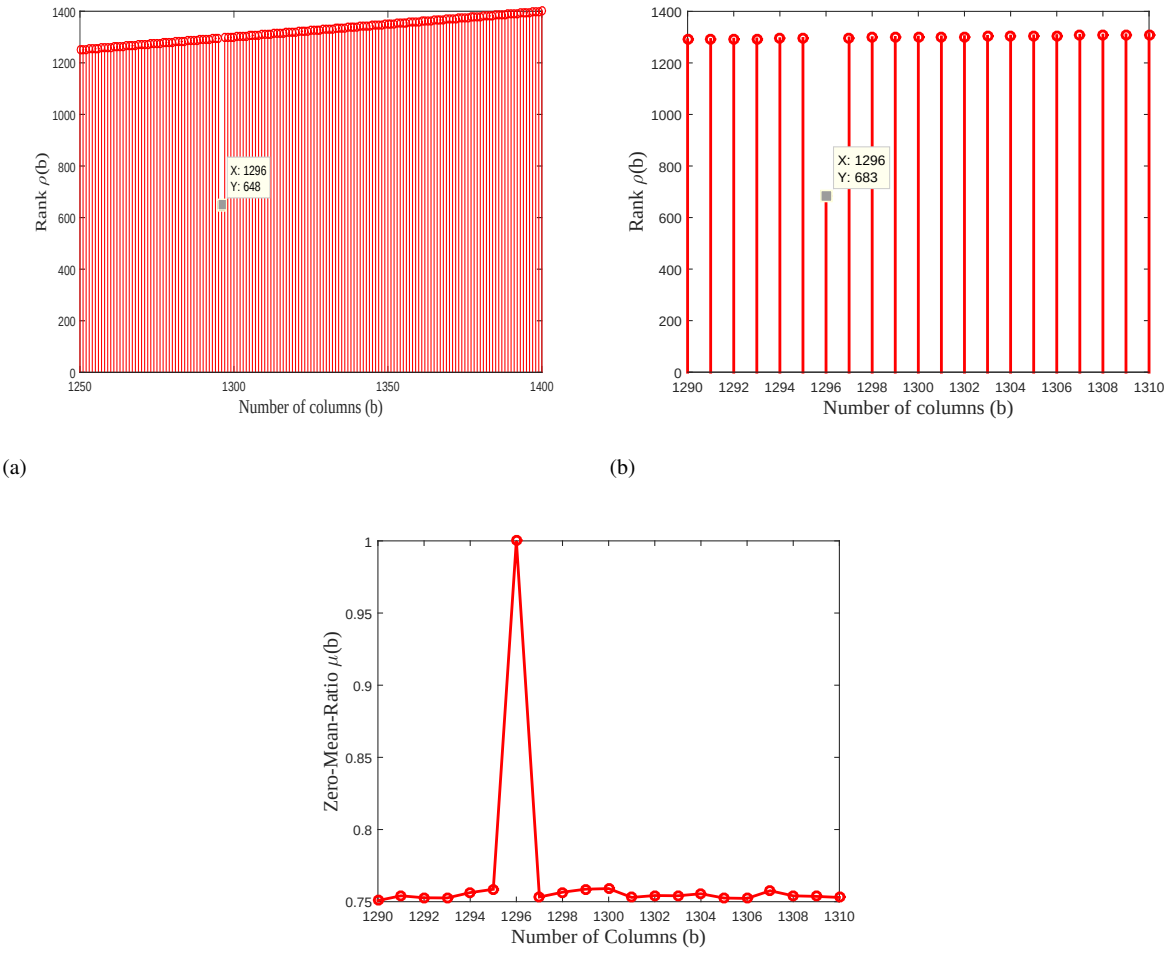
SNR (dB)	$r = 5/6$, $M = 2$	$r = 1/4$, $M = 2$	$r = 5/6$, $M = 4$	$r = 1/4$, $M = 4$
12	108 (100%)	486 (100%)	108 (100%)	486 (100%)
11	108 (100%)	486 (100%)	102 (94.44%)	474 (97.5%)
10	103 (95.37%)	485 (99.79%)	0 (0%)	122 (25.10%)
9	24 (22.22%)	341 (70.16%)	0 (0%)	0 (0%)
8	0 (0%)	10 (2.05%)	0 (0%)	0 (0%)
7	0 (0%)	0 (0%)	0 (0%)	0 (0%)

Table 3 Accuracy of estimation of dependent columns $n - k$ based on Algorithm 1 assuming $n = 1296$

SNR (dB)	$r = 5/6$, $M = 2$	$r = 1/4$, $M = 2$	$r = 5/6$, $M = 4$	$r = 1/4$, $M = 4$
12	216 (100%)	972 (100%)	216 (100%)	972 (100%)
11	216 (100%)	972 (100%)	184 (85.18%)	934 (96.09%)
10	202 (93.51%)	962 (98.97%)	0 (0%)	34 (3.4%)
9	4 (1.8%)	463 (47.63%)	0 (0%)	0 (0%)
8	0 (0%)	0 (0%)	0 (0%)	0 (0%)

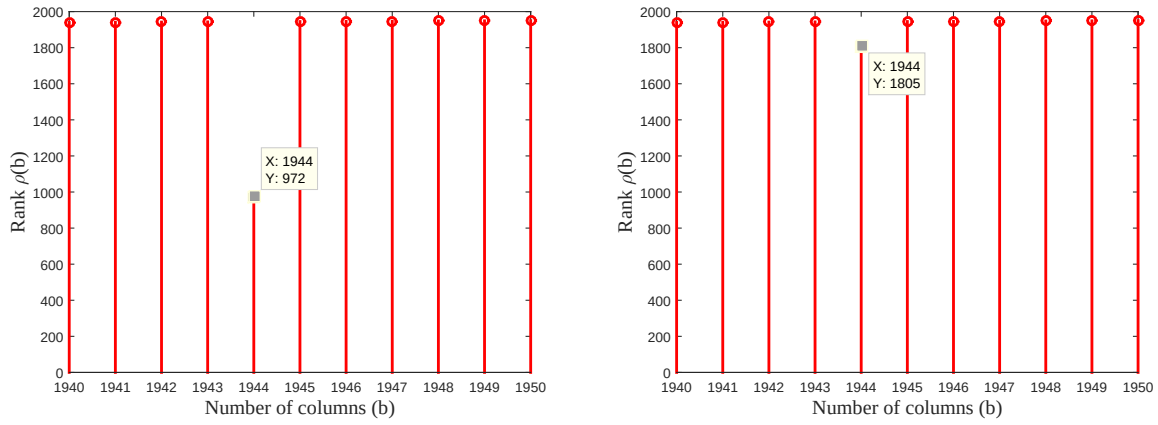
is observed that both n and k can be estimated with 100% accuracy for BPSK and QPSK schemes when SNR $\geq$ 11 dB and 12 dB, respectively. If the number of dependent columns is estimated with less than 100% and more than 0% accuracy, then atleast n alone can be successfully recognized. With this fact, it is inferred from Table 2 that for $r = \frac{5}{6}$ case, n can be estimated correctly for BPSK and QPSK schemes when SNR $\geq$ 9 and 11 dB, respectively. For the case when $r = \frac{1}{4}$, n can be estimated correctly for BPSK and QPSK schemes when SNR $\geq$ 8 and 10 dB, respectively.

It is observed from Table 3 and Table 4 that both n and k can be estimated with 100% accuracy for BPSK and QPSK schemes when SNR $\geq$ 11 and 12 dB, respectively. It is also inferred from Table 3 and Table 4 that n alone can be estimated correctly for BPSK and QPSK schemes when SNR $\geq$ 9 and 11 dB, respectively,



(a) (b) (c)

Fig. 6: *(a) Rank Vs Number of columns considering $r = 1/2$, $n = 1296$, $k = 648$, $M = 4$, SNR = 12 dB, and $a = 2 \cdot b$ (b) Rank Vs Number of columns considering $r = 1/2$, $n = 1296$, $k = 648$, $M = 4$, SNR = 11 dB, and $a = 2 \cdot b$ (c) Zero-mean-ratio Vs Number of columns considering $r = 1/2$, $n = 1296$, $k = 648$, $M = 4$, SNR = 10 dB, and $a = 2 \cdot b$*

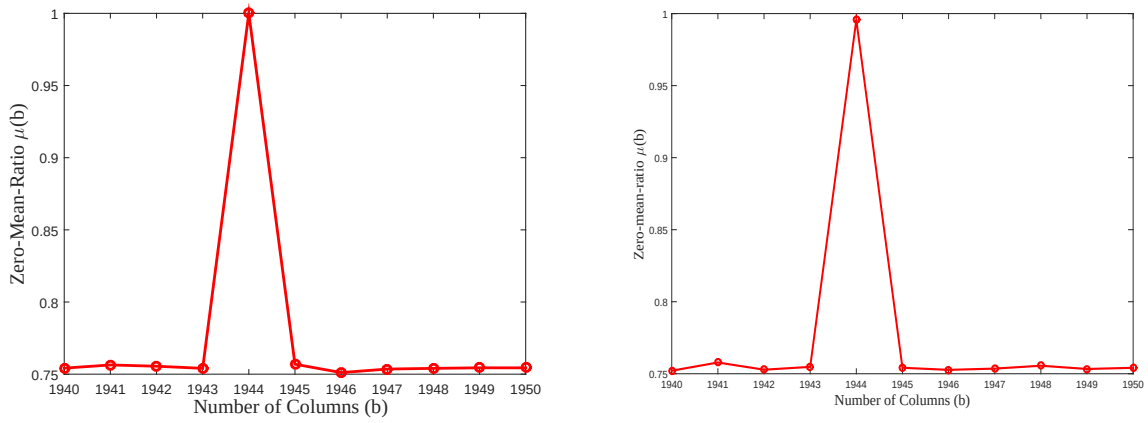


(a) (b)

Fig. 7: *(a) Rank Vs Number of columns considering $r = 1/2$, $n = 1944$, $k = 972$, $M = 2$, SNR = 11 dB, and $a = 2 \cdot b$ (b) Rank Vs Number of columns considering $r = 1/2$, $n = 1944$, $k = 972$, $M = 2$, SNR = 9 dB, and $a = 2 \cdot b$*

considering $r = \frac{5}{6}$ case. It is noticed that the performance trend for $n = 1944$ case is similar to $n = 1296$ case with lesser accuracy. For the case when $r = \frac{1}{4}$, $n = 1296$ can be estimated correctly for BPSK and QPSK schemes when SNR $\geq$ 9 and 10 dB, respectively, with better accuracy compared to $n = 1944$ case.

In a nutshell, it is inferred from Table 2, Table 3, and Table 4 that the accuracy of estimation improves with decrease in modulation order M, as expected and the accuracy deteriorates with increase in r and n. As r increases, the number of dependent columns (i.e. $n - k$) decreases. Hence, it is difficult to classify rank-deficient and

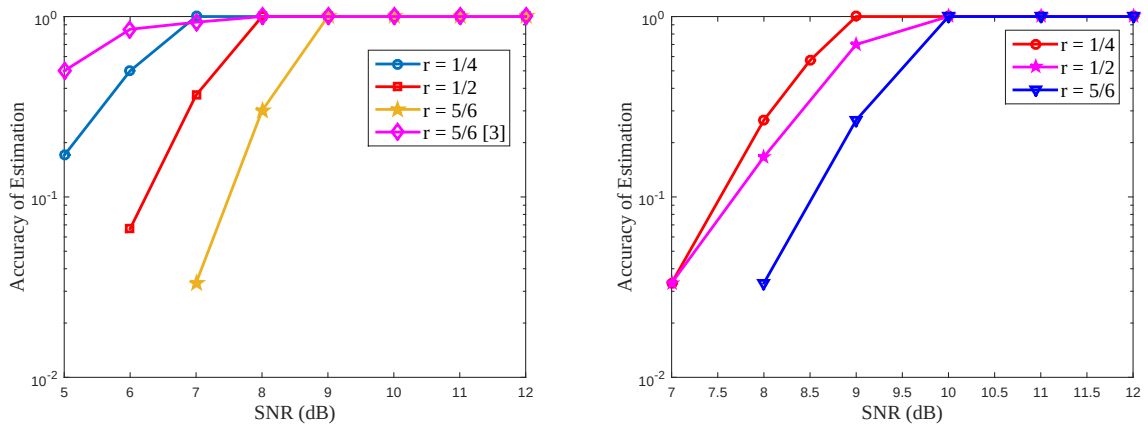


(a) (b)

Fig. 8: *(a) Zero-mean-ratio Vs Number of columns considering $r = 1/2$, $n = 1944$, $k = 972$, $M = 2$, SNR = 9 dB, and $a = 2 \cdot b$ (b) Zero-mean-ratio Vs Number of columns considering $r = 1/2$, $n = 1944$, $k = 972$, $M = 2$, SNR = 8 dB, and $a = 2 \cdot b$*

Table 4 Accuracy of estimation of dependent columns $n - k$ based on Algorithm 1 assuming $n = 1944$

SNR (dB)	$r = 5/6$, $M = 2$	$r = 1/4$, $M = 2$	$r = 5/6$, $M = 4$	$r = 1/4$, $M = 4$
12	324 (100%)	1458 (100%)	324 (100%)	1458 (100%)
11	324 (100%)	1458 (100%)	264 (81.48%)	1385 (94.99%)
10	303 (93.51%)	1422 (97.53%)	0 (0%)	11 (0.7%)
9	2 (0.6%)	503 (34.49%)	0 (0%)	0 (0%)
8	0 (0%)	0 (0%)	0 (0%)	0 (0%)



(a) (b)

Fig. 9: *(a) Accuracy of estimation of codeword length $n = 648$ using Algorithm 2 assuming $M = 2$ for different code rate values and comparison with an existing method [3] (b) Accuracy of estimation of codeword length $n = 648$ using Algorithm 2 assuming $M = 4$ for different code rate values*

full rank data matrices for bad propagation environment based on $\rho(b)$ due to less number of dependent columns.

In Fig. 9(a) and Fig. 9(b), the accuracy of estimation of codeword length $n = 648$ based on Algorithm 2 is shown for different values of code rate r considering BPSK and QPSK schemes, respectively. From Fig. 9(a), it is observed that the accuracy of estimation is 100% when SNR $\geq$ 7 dB, 8 dB, and 9 dB for $r = 1/4$, $1/2$, and $5/6$, respectively. Similarly, it is noticed from Fig. 9(b) that the accuracy of estimation is 100% when SNR $\geq$ 9 dB for $r = 1/4$ and 10 dB for r =1/2 and $5/6$. Thus, it is inferred from both the curves that the accuracy of estimation deteriorates with increase in code rate r and modulation order M similar to Algorithm 1. Moreover, improvement in the accuracy of estimation with SNR gain of atleast 1 dB in estimating $n = 648$ is observed using Algorithm 2 compared to Algorithm 1 (refer to Table 2). Since Algorithm 2 estimates n by increasing the number of iterations i.e. $N = 10$, the computational time also increases. It is to be noted that when $N = N_{\text{frame}} - a + 1 > 10$, where N_{frame} denotes the number of frames (refer to Algorithm 2), further improvement in the accuracy of estimation of about 1 dB is observed with increase in the computational time compared to $N = 10$ case. The main reason behind increase in the computational time is due to Gauss elimination process and the number of Gauss eliminations increases with the number of iterations N. For example, the number of Gauss eliminations required to evaluate rank $\rho(b)$ in Algorithm 1 when b is varied from b_{min} to b_{max} is $b_{\text{max}} - b_{\text{min}}$. Similarly, the number of Gauss eliminations required to evaluate zero-mean-ratio $\mu(b)$ in Algorithm 2 is $N \cdot (b_{\text{max}} - b_{\text{min}})$. The Gauss elimination process in both Algorithm 1 and 2 operates on a data matrix of size $a \times b$, where $a = 2 \cdot b$. Note that the Gauss elimination process in Algorithm 2 operates on a data matrix with increase in the number of iterations compared to Algorithm 1. Therefore, the computational

time is one of the important challenges to achieve better performance in the case of blind estimation of LDPC code parameters assuming large codeword length values.

We have also compared the proposed algorithm against the available literature. A novel method for identifying LDPC code parameters is proposed in [17]. However, the proposed method is limited to noiseless scenario and is suitable only for moderate codeword lengths. In [3], the identification of true LDPC encoder is carried out for BPSK modulation scheme based on the average log-likelihood ratio (LLR) of syndrome a posteriori probability (SPP). A pre-defined candidate set, which contains finite number of LDPC encoders, is assumed to be known to both transmitter and receiver and the average LLR of SPP is evaluated for the candidate set. The corresponding encoder with maximum average LLR value is chosen as the true LDPC encoder. The proposed algorithm is compared with the existing method in Fig. 9(a) for rate $r=5/6$ case assuming $n=648$ and BPSK modulation scheme. It is noticed that the accuracy of estimation is 100% when SNR $\geq$ 8 dB and 9 dB for the existing LLR-based method and the proposed algorithm, respectively. Therefore, the existing LLR method in [3] performs better in terms of probability of correct detection of true LDPC encoder compared to the proposed algorithm. However, as already mentioned in Section 1.1, the LLR-based estimation of true LDPC encoder is suitable only for adaptive modulation and coding (AMC) systems, since a predefined LDPC encoder candidate set is assumed to be known at the transmitter and receiver. In addition, the LLR-based method is not suitable for blind code parameter estimation, which is essential for non-cooperative military and spectrum surveillance systems. On the other hand, the proposed algorithm is a more generic one covering both cooperative (i.e. AMC-based) and non-cooperative systems. In the proposed algorithm, the candidate set is unknown and that necessitates the blind estimation of LDPC code parameters. Similarly, the proposed algorithm to estimate codeword length and code dimension of LDPC codes can be combined with the LLR-based method proposed in [3] to estimate the parity check matrix H and the same has been discussed briefly in Section 4.4.

4.3 Extension to non-binary codes

The proposed algorithms can be extended to non-binary LDPC codes, which are constructed based on Reed-Solomon (RS) codes, as follows: Firstly, the RS code with symbols from Galois field $GF(2^m)$ has the following parameters [31]:

- Codeword length $n=2^m-1$
- Number of parity check symbols $n-k=2t$
- Code dimension $k=2^m-1-2t$
- Minimum hamming distance $d_{\text{min}}=2t+1$

After demodulation at the receiver, the incoming binary symbols will be converted into non-binary symbols while simultaneously varying the number of bits per symbol m, where $m \in [m_{\text{min}}, m_{\text{max}}]$ and primitive polynomials $p(x)$ corresponding to m. A Galois field (GF) array is created after converting the binary coded data symbols into non-binary symbols between 0 and 2^m-1. This array interprets the integers between 0 and 2^m-1 with respect to a specific primitive polynomial for that field. Now the GF array with non-binary LDPC coded symbols is reshaped into a matrix S of size $a \times b$, where a and b denote the number of rows and columns of data matrix S, respectively, and $b=2^m-1$. The rank and rank ratio of S are calculated using finite-field Gauss elimination process [24] instead of Gauss elimination process. Since non-binary LDPC coded symbols belong to finite-field or GF, the rank and rank ratio are calculated using finite-field Gauss elimination method. From the rank ratio values, the corresponding combination of $[m,p]$, which minimizes the rank ratio is chosen as the estimated non-binary LDPC code parameters m_{est} and p_{est} [16]. It is to be noted that p denotes the integer representation of primitive polynomial $p(x)$. Since codeword length $n=2^m-1$ for non-binary LDPC code constructed using RS code, n_{est} is obtained as follows: $n_{\text{est}}=2^{m\text{est}}-1$. Similar to binary LDPC codes, the rank value corresponding to n_{est} will give the code dimension k_{est}. In case of more erroneous channel conditions, the zero-mean-ratio values are calculated instead of rank ratio for different values of $[m,p]$ for N iterations similar to Algorithm 2. The corresponding combination which maximizes the zero-mean-ratio is chosen as m_{est}, n_{est}, and p_{est}, where $n_{\text{est}}=2^{m\text{est}}-1$.

However, the proposed algorithms can also be extended to shortened non-binary LDPC codes (i.e. $n<2^m-1$) with minor modification as follows: The corresponding combination of $[m,p,b]$, where $b \in [b_{\text{min}}, b_{\text{max}}]$, which minimizes the rank ratio or maximizes zero-mean-ratio is chosen as the estimated non-binary LDPC code parameters $[m_{\text{est}}, p_{\text{est}}, n_{\text{est}}]$, respectively.

4.4 Discussion on the estimation of parity check matrix

The parity check matrix i.e. H matrix and the code dimension can be successfully estimated only when the received data is synchronized. To estimate the starting position of the received data symbols, the proposed algorithm for less erroneous channel conditions i.e. Algorithm 1 is modified as follows: After estimating n_{est}, we consider a sliding synchronization window of length n_{est}. The rank is calculated by sliding the window across the data symbols using bit position adjustment parameter ϕ from 0 to $n_{\text{est}}-1$. Now the corresponding value of ϕ, which minimizes the rank is chosen as the estimate of bit position adjustment parameter ϕ_{est}. The rank value corresponding to ϕ_{est} gives the estimate of code dimension k_{est}. Similarly, to estimate the starting position of the received data symbols, the proposed algorithm for more erroneous channel conditions i.e. Algorithm 2 is modified as follows: After estimating n_{est}, the rank is calculated by sliding the synchronization window across the data symbols using bit position adjustment parameter ϕ from 0 to $n_{\text{est}}-1$. Now the corresponding value of ϕ, which maximizes the zero-mean-ratio will be chosen as ϕ_{est}. After shifting ϕ_{est} bit positions, synchronization of the received data sequence is achieved. After synchronization, the possibility of estimating true H matrix and its challenges are discussed below.

The H matrix of LDPC code can be identified in two possible ways. The first method is based on the identification of dual codewords and the second method is based on the soft-decision outputs i.e. average LLR of SPP. The first method has been explained as follows: The data matrix S is converted into its column echelon form F using modified Gauss-Jordan elimination through pivoting (GJETP) algorithm as follows $S.\chi=F$, where χ denotes the column permutation matrix. The columns in χ corresponding to all-zero columns in F are called dual-codes. If $b=n_{\text{est}}$, then there exist $n-k$ all-zero columns in F for non-erroneous or less erroneous channel conditions (refer Table 2 to 4) according to (1). Therefore, the subspace spanned by $n-k$ column vectors in χ corresponding to $n-k$ all-zero columns in F is called dual-code subspace. The parity check matrix H with dimension $(n-k)\times n$, which characterizes the dual code subspace, can be identified by taking the transpose of the matrix with $n-k$ column vectors in χ. For more erroneous channel conditions, the columns in χ corresponding to the columns in F with zero-mean-ratio values lesser than a particular threshold value are selected as dual-codes.

An alternate method based on LLR of SPP is also suggested. Since LDPC encoder with large codeword length will have a very large parity check matrix H, it is difficult to blindly reproduce the exact H matrix without any apriori knowledge. Note that the estimation of H for any linear block code can be performed by generating possible H matrices for estimated values of n and k. Among the possible matrices, the true or correct H matrix should satisfy the condition $H \cdot c^{\top}=0$, where c denotes the codeword. However, this method is valid only for less erroneous channel conditions. For more erroneous channel conditions, the condition should be changed appropriately and the likelihood of a particular H matrix satisfying the condition $H \cdot c^{\top}=0$ will be identified based on the LLR values [4]. As n is large for LDPC codes, the number of possible H matrices that can be generated will also be large. It is a very tedious process to evaluate LLR values for all possible H matrices. Hence, a

pre-defined candidate set, which contains finite number of H matrices for the estimated value of n, is assumed to be known to both transmitter and receiver. Now the LLR of SPP is evaluated for the candidate set and the corresponding H matrix for which the LLR of SPP is maximum can be chosen as the true H matrix assumed in the transmitter. The candidate set assumed in [3], which contains various LDPC encoder candidates, is constructed without any apriori knowledge about the code parameters. However, the candidate set suggested in our work, which contains finite number of possible H matrices, can be constructed based on the estimated value of n using the proposed blind estimation algorithms. Hence, the number of H matrices in the candidate set will be lesser than the candidate set assumed in [3]. In a nutshell, the LLR calculation to estimate true H matrix is not very complicated, if a pre-defined candidate set with finite number of H matrices is assumed to be known.

5 Conclusion

In this paper, we propose algorithms for the blind estimation of codeword length and code dimension of LDPC codes considering a non-cooperative scenario. The algorithms are proposed for less and more erroneous channel conditions based on deficient rank and zero-mean-ratio values, respectively. Test cases are given for three different codeword length values i.e. $n=648$, 1296, and 1944 for $M-$PSK schemes. From the simulation results, it is observed that the accuracy of estimation of code parameters deteriorates with increase in modulation order, code rate, and codeword length values. Further, performance improvement of atleast 2 dB in estimating n is observed based on Algorithm 2 compared to Algorithm 1 with increase in the computational time. Hence, it is concluded that the blind estimation of LDPC codes considering large codeword lengths is a challenging process in terms of computational time. Finally, the extension of the proposed algorithms to non-binary RS codes and possible methods to estimate H matrix of LDPC codes are also discussed.

6 References

[1] Ziegler, J. F.: 'Automatic Recognition and Classification of Forward Error Correcting Codes'. M.S. thesis, Dept. Elect. Comput. Eng., George Mason Univ., Fairfax, VA, USA, 2000.

[2] Zhang, H., Wu, H-C., Jiang, H.: 'Novel blind encoder identification of Reed-Solomon codes with low computational complexity', Proc. IEEE GLOBECOM, Atlanta, USA, December 2013, pp. 3294–3299.

[3] Xia, T., Wu, H-C.: 'Novel blind identification of LDPC codes using average LLR of syndrome a posteriori probability', *IEEE Trans. Signal Process.*, 2014, **62**, (3), pp. 632–640.

[4] Moosavi, R., Larsson, E. G.: 'Fast blind recognition of channel codes', *IEEE Trans. Commun.*, 2014, **62**, (5), pp. 1393–1405.

[5] Marazin, M., Gautier, R., Burel, G.: 'Dual code method for blind recognition of convolutional encoder for cognitive radio receiver design', Proc. IEEE GLOBECOM, Honolulu, Hawaii, USA, December 2009, pp. 1–6.

[6] Marazin, M., Gautier, R., Burel, G.: 'Blind recovery of k/n rate convolutional encoders in a noisy environment', *EURASIP J. Wirel. Commun. and Netw.*, 2011, **2011:168**, pp. 1–9.

[7] Rosen, G. L.: 'Examining coding structure and redundancy in DNA', *IEEE Eng. Med. Biol. Mag.*, 2006, **25**, (1), pp. 62–68.

[8] Tillich, J-P., Tixier, A., Sendrier, N.: 'Recovering the interleaver of an unknown Turbo-Code', Proc. IEEE ISIT, Honolulu, Hawaii, USA, July 2014, pp. 2784–2788.

[9] Dingel, J., Hagenauer, J.: 'Parameter estimation of a convolutional encoder from noisy observations', Proc. IEEE ISIT, Nice, France, June 2007, pp. 1776–1780.

[10] Cluzeau, M., Finiasz, M.: 'Reconstruction of punctured convolutional codes', Proc. IEEE ITW, Taormina, Dec. 2009, pp. 75–79.

[11] Marazin, M., Gautier, R., Burel, G.: 'Algebraic method for blind recovery of punctured convolutional encoders from an erroneous bit stream', *IET Signal Process.*, 2012, **6**, (2), pp. 122–131.

[12] Marazin, M., Gautier, R., Burel, G.: 'Some interesting dual-code properties of convolutional encoder for standards self-recognition', *IET Commun.*, 2012, **6**, (8), pp. 931–935.

[13] Jing, Z., Zhiping, H., Shaojing, S., Shaowu, Y.: 'Blind recognition of binary cyclic codes', *EURASIP J. Wirel. Commun. Netw.*, 2013, **2013:218**, pp. 1–17.

[14] Swaminathan, R., Madhukumar, A. S.: 'Classification of error correction codes and estimation of interleaver parameters in a robust environment', *IEEE Trans. Broadcast.*, 2017, **63**, (3), pp. 463–478.

[15] Zrelli, Y., Marazin, M., Gautier, R., Rannou, E., Radoi, E.: 'Blind identification of code word length for non-binary error-correcting codes in noisy transmission', *EURASIP J. Wirel. Commun. Netw.*, 2015, **2015:43**, pp. 1–16.

[16] Swaminathan R., Madhukumar, A. S., Wang, G., Kee, T. S.: 'Blind reconstruction of Reed-Solomon encoder and interleavers over noisy environment', *IEEE Trans. Broadcast.*, Early access, **99**, (PP), pp. 1–16.

[17] Refaey, A., Niati, R., Wang, X., Chouinard, J. Y-: 'Blind detection approach for LDPC, convolutional, and turbo codes in non-noisy environment', Proc. IEEE Conference on Communications and Network Security, San Francisco, CA, USA, Oct. 2014, pp. 502–503.

[18] Moosavi, R., Larsson, E. G.: 'A fast scheme for blind identification of channel codes', Proc. IEEE GLOBECOM, Texas, USA, Dec. 2011, pp. 1–5.

[19] Xia, T., Wu, H-C.:, 'Novel blind identification of LDPC codes using average LLR of syndrome a posteriori probability', Proc. IEEE International Conference on Telecommunications (ITS), Taipei, Nov. 2012, pp. 12–16.

[20] Yu, P., Peng, H., Li, J.: 'On blind recognition of channel codes within a candidate set', *IEEE Commun. Lett.*, 2016, **20**, (4), pp. 736–739.

[21] Debessu, Y. G., Wu, H.-C., Jiang, H.: 'Novel Blind Encoder Parameter Estimation for Turbo Codes', *IEEE Commun. Lett.*, 2012, **16**, (12), pp. 1917–1920.

[22] Yu, P., Li, J., Peng, H.: 'A least square method for parameter estimation of RSC sub-codes of turbo codes', *IEEE Commun. Lett.*, 2014, **18**, (4), pp. 644–647.

[23] Sicot, G., Houcke, S., Barbier, J.: 'Blind detection of interleaver parameters', *Signal Process.*, 2009, **89**, pp. 450–462.

[24] Lu, L., Li, K. H., Guan, Y. L.: 'Blind detection of interleaver parameters for non-binary coded data streams', Proc. IEEE ICC, Dresden, Germany, June 2009, pp. 1–4.

[25] Swaminathan, R., Madhukumar, A. S., Teck, N. W., Samson, S. C. M.: 'Parameter estimation of block and helical scan interleavers in the presence of bit errors', *Digital Signal Process.*, 2017, **60**, pp. 20–32.

[26] Swaminathan, R., Madhukumar, A. S.: 'Joint recognition of error correcting codes and interleaver parameters in a robust environment', Proc. IEEE PIMRC, Valencia, Spain, September 2016, pp. 1–6.

[27] Lu, L., Li, K. H., Guan, Y. L.: 'Blind identification of convolutional interleaver parameters', Proc. IEEE ICICS, Macau, China, December 2009, pp. 1–5.

[28] Jia, Y-Q., Li, L-P., Li, Y-Z., Gan, L.: 'Blind estimation of convolutional interleaver parameters', Proc. IEEE WiCOM, Shanghai, China, September 2012, pp. 1–4.

[29] Swaminathan, R., Madhukumar, A. S., Teck, N. W., Samson, S. C. M.: 'Parameter estimation of convolutional and helical interleavers in a noisy environment', *IEEE Access*, 2017, **5**, pp. 6151–6167.

[30] Choi, C., Yoon, D.: 'Enhanced blind interleaver parameters estimation algorithm for noisy environment', *IEEE Access*, Early access, **PP**, (99), pp. 1–6.

[31] Lin, S., Costello, D. J.: 'Error Control Coding', (Prentice hall publication, 2004, 2nd edn.).